Software Requirements Specification (SRS) Document

Team NaClStack

Shreyas Badami, Kapil Rajesh Kavitha

Overview

The human genome consists of billions of base pairs. There is a need for a genome browser to enable genomic researchers to visualize and explore the genetic information stored in a genome using a user-friendly, palatable interface. This project aims to create a FASTA file browser along with other tools which would help genomic researchers.

The browser would also contain tools to analyze the genomic variations between people (for example, in facial features) from different countries/populations, by grouping the genes affecting the same part of the body together. A tool to implement variation analysis from a genome-level standpoint would also be implemented.

System Requirements

Functional requirements (described using use cases)

1. Frontend (User interface)
    a. Basic UI: React-based application user interface built for uploading data and viewing results.
    b. Input form: A form to take input from the user, which will be used to specify:
        i. Search mode to be used; BLAST or name based
        ii. Tool to be used
        iii. Various input parameters such as algorithm to be used and scoring matrices.
    c. Upload sequences: Upload a FASTA file from device which contains sequences, which will be sent to the backend for processing.
    d. Data display: Display the processed output according to the tool applied on it
        i. Alignment tool – shows a color-coded alignment of the input sequences with all the indels and differences highlighted.
        ii. Phylogenetic tree generator – creates a phylogenetic tree out of the input sequences provided, indicating how closely the various sequences are related.
        iii. Variation analyzer – to display which variation a gene belongs to, as well as to show the difference between that gene and the reference genome.
2. Backend
    a. Express backend – to be accessed using API calls sent from the frontend, or direct API calls. Will be connected to the database provided.
        i. Storing input sequences: input sequences will be stored and passed on to the Flask backend for processing
        ii. Search tool: implement a fuzzy search tool to return similarly named FASTA files.
    b. Python Backend – to be accessed by the Express backend using FAST API calls. The tools used for processing the input sequences are implemented here.
        i. Implementing the BLAST algorithm to find sequences similar to the input sequence.
        ii. Implement various pairwise alignment algorithms such as Needleman-Wunsch, Waterman-Smith etc.
        iii. Implement various multiple sequence alignment algorithms such as Clustal W, MUSCLE etc.
        iv. Implement the phylogenetic tree generator.
        v. Implement the alignment viewer tool.
        vi. Implement the variation analysis tool

Non-functional requirements

- Usability requirements: To work on all devices/browsers that can access the browser and be able to provide FASTA files as inputs. The tool should also be user-friendly.
- Performance-related requirements: Response time of <2 minutes for a given input.
- Scalability related requirements:
- Security related requirements: Only allow authenticated genomic researchers to access the browser. The researchers should be able to access only the relevant parts of the genome

Project Deliverables

- FASTA browser which supports 2 kinds of search; BLAST based similarity search and a simple name-based search.
- A set of tools which work on the database:
    - A sequence alignment tool which implements both pairwise alignment and multiple sequence alignment.
    - A color coded alignment viewer, which shows the indels and SNPs present.

- A variation analyzer, which would aim to analyze genomic variations between people from different groups.

Examples

2 sequences aligned using EMBOSS

```
EMBOSS_001         1 ATGAAATACAAAGCCCTGCCCTTACTGCCGCTTGCCGCCGCCCTTGCCGC     50
                     ||||||||||||||||||.||||||||||||||||||.||||||||||||
EMBOSS_001         1 ATGAAATACAAAGCCCTGTCCTTACTGCCGCTTGCCGCTGCCCTTGCCGC     50

EMBOSS_001        51 CTGTGCCGGGGGGGGGGGTAGCCGAACCGCACGTCCCCGTGTCCATCCCCA    100
                     ||||||||||||||||||||||||||||||||||||.|.|||||.||||
EMBOSS_001        51 CTGTGCCGGGGGGGGGGGTAGCCGAACCGCACGTCCCCTTCTCCATTCCCA    100

EMBOSS_001       101 CCGCCACGCCGCTGCCCGCCGGCGAGGTAACGTTATCAAGCGATAACGGC    150
                     ||||||||||||||   .|||||||||||||||||||||.|||||.||..
EMBOSS_001       101 CCGCCACGCCGCTG---ACCGGCGAGGTAACGTTATCAACCGATAGCGCA    147

EMBOSS_001       151 AATATCGAAAACATCAACACCGCCGGCGCCGGAAGCGCATCCGACGCGCC    200
                     ||.|||||||||||||.|||||||||.|.|||||||.|||
EMBOSS_001       148 AACATCGAAAACATCAATACCGCCGGCACAGGAAGCACAT----------    187

EMBOSS_001       201 GAGCCGCAGCAGACGCTCGCTCGATGCCGCCCCGCAAAACACATCCGGCA    250
                                                         |||||||||
EMBOSS_001       188 ----------------------------------------CATCCGGCA    196
```

| -> match
. -> SNP
- -> indel